# 決定論的 2 階パターンとプログラム変換への応用

Deterministic Second-order Patterns in Program Transformation

横山 哲郎 † 胡 振江 †,†† 武市 正人 †

Tetsuo YOKOYAMA Zhenjiang HU Masato TAKEICHI

† 東京大学大学院情報理工学系研究科

Graduate School of Information Science and Technology, University of Tokyo

†† 科学技術振興事業団 さきがけ 21

PRESTO21, Japan Science and Technology Corporation

yokoyama@ipl.t.u-tokyo.ac.jp, {hu,takeichi}@mist.i.u-tokyo.ac.jp

2 階パターンと 2 階マッチングを用いると高度なプログラム変換を記述することができることが知られている [4]．しかし 2 階マッチングは NP 完全であるため効率がよい実装が望めず，また非決定論的であるため，セマンティクスが複雑であるという問題点がある．われわれは，2 階パターンの形を制限することで，どのような項とも高々1 つのマッチしか得られないようなパターンのクラスを決定論的 2 階パターンとして定め，またこのマッチを得るための効率の良いアルゴリズムを開発した [6]．本論文では，このような決定論的 2 階パターンのクラスを拡張し，また決定論的線形 2 階パターンであるための必要十分条件を与える．このパターンのクラスを用いる幅広い応用が可能であると考えられる．

## 1 はじめに

2 階パターンと 2 階マッチングを用いて記述するプログラム変換は記述力が高いことが知られており [4]，実際に，いくつかのプログラム変換システムに実装されている [2, 3]．

しかし，2 階マッチングアルゴリズムは NP 完全である [1] ため効率の良いアルゴリズムの実装は望めず，また非決定論的 (nondeterministic) であるという欠点があり，関数プログラムで直接的に扱うことはできないしセマンティクスが複雑になるため望ましくない．

われわれは，2 階パターンの形を制限することで，どのような項とも高々 1 つのマッチしか返さないパターンのクラスを定めた [6]．このクラスは決定論的単一化の文脈で出てくる Miller の決定論的高階パターンのクラス [5] の拡張である．この結果，2 階パターンと 2 階マッチングによる記述力の高さをもったプログラム変換の仕様記述が可能になった．

本論文では，文献 [6] の決定論的 2 階パターンの十分条件を拡張する．また，この拡張の限界を示す決定論的線形 2 階パターンの必要十分条件を与える．このパターンはプログラム変換の実装や関数プログラムのパターンの拡張において活用することができると考えられる．

## 2 決定論的 2 階パターン

決定論的 2 階パターンを定義するために，準備としていくつかの用語を説明する．

われわれは，定数，変数，適用，$\lambda$ 抽象によって再帰的に定義される通常の単純型付け $\lambda$ 項を考える．

$$T \;=\; c \mid v \mid T\ T \mid \lambda x.T$$

$FV$ を項からその項中の自由変数の集合への関数とする．$Args$ を項からその項の $\lambda$ 抽象の引数の集合への関数とする．項 $E$ が自由変数を持たないとき，項 $E$ は閉じているという．読みやすさのため，中置演算子を用いる．$x + y$ は $(+)\ x\ y$ を意味している．項が $\beta$-redex を含まないとき $\beta$ 正規形と呼ぶ．項が $\eta$-redex を含まないとき $\eta$ 正規形（または $\eta$-short 正規形）と呼ぶ．読みやすさのため，ときどき $\lambda x_1 \cdots x_l.p\ E_1 \cdots E_m$ を $\lambda \bar{x}.p\ \bar{E}$ と書く．文脈は $C[\_], D[\_,\_]$ などで書く．文脈は自由変数を含まないことにし，引数は必ず文脈中に出現することにする．

$E_1 \in subTerm\ E_2$ のとき，項 $E_1$ は $E_2$ の部分項であるといい，$E_1 \trianglelefteq E_2$ と書く．ここで $subTerm$ は図 1のように定義されている関数である．$v\ t_1\ \ldots\ t_n$ を部分項とし $v$ を変数としたとき，$t_1, \ldots, t_n$ を $v$ の引数と呼び，$v$ をこの部分項の頭と呼ぶ．

置換は変数から項への写像として例えば次のよう

$$
\begin{array}{lcl}
subTerm(c) & = & \{c\} \\
subTerm(v) & = & \{v\} \\
subTerm(E_1\ E_2) & = & \{E_1\ E_2\} \cup subTerm(E_1) \cup \\
 & & subTerm(E_2) \\
subTerm(\lambda x.\, E) & = & \{\lambda x.\, E\} \cup subTerm(E)
\end{array}
$$

図 1: 関数 $subTerm$

に書かれる．

$$\phi = \{p \mapsto \lambda x.\, x\ b\}$$

マッチング，単一化で得られる置換をそれぞれマッチ，単一化子と呼ぶ．特に，マッチの値域は閉じた項である．置換 $\phi$ の定義域を $dom(\phi)$，値域を $range(\phi)$ と書く．置換 $\phi, \psi$ が条件:

$$\forall v \in dom(\phi) \cap dom(\psi)\,.\, \phi\ v =_{\alpha\beta\eta} \psi\ v$$

を満たすとき compatible であるという．ここで，等号 $(=_{\alpha\beta\eta})$ は $\alpha\beta\eta$ 変換における同値関係を表している．

型は単純型付け $\lambda$ 式で通常通り構成される．$T_0$ を基底型とする．型 $T$ は次のように定義される．

1. $\alpha \in T_0$ のとき $\alpha \in T_0$ である．
2. $\alpha, \beta \in T_0$ のとき $\alpha \to \beta \in T_0$ である．
3. $T$ を (1),(2) 以外のものを含まない．

型 $\tau$ のオーダー $ord(\tau)$ は次のように定義される．

$$
\begin{array}{lcll}
ord(\alpha) & = & 1, & \textbf{if } (\alpha \in T_0) \\
ord(\alpha \to \beta) & = & max\{ord(\alpha) + 1, ord(\beta)\} &
\end{array}
$$

基底型のオーダーは 1 である．関数型のオーダーは引数の型に 1 を足したものと返す型の大きいものである．項のオーダーはその型のオーダーである．パターンのオーダーはパターン中に出現する自由変数の型の最大値である．

自由変数を持つ項をパターンという．パターン中で同一の自由変数が 1 回しか出現しないパターンを線形パターンという．

$\beta\eta$ 正規形のパターン $P$ と閉じた項 $T$ の組 $P \to T$ をルールという．パターン $P$ と閉じた項 $T$ を用いて $\phi\ P =_{\alpha\beta\eta} T$ であることを $\phi \vdash P \to T$ と書き，この置換 $\phi$ をもとめることをマッチングといい，置換 $\phi$ をマッチという．項 $P, Q$ と置換 $\phi$ を用いて，$\phi\ P =_{\alpha\beta\eta} \phi\ Q$ であることを $\phi \vdash P \sim Q$ と書く．

決定論的パターンとはどのような項をもってきても多くても 1 つの最汎マッチしか返さないようなパターンのことを言う．われわれのパターンは決定論的単一化の文脈で出てくる Miller の決定論的パターン [5] の拡張である．まずは Miller の高階パターンを 2 階の $\lambda$ 式に制限した次のようなパターンを考える．

**定義 1 (制限された Miller の 2 階パターン)** パターン中の自由変数の引数は互いに異なる束縛変数である 2 階パターン $P$ を制限された Miller の 2 階パターンと呼ぶ． □

例えば，$\lambda x\, y.\, p\ y\ x$ は Miller の高階パターンである．なぜなら，$p$ は 2 階の自由変数であり $x, y$ は互いに異なる束縛変数あるからである．以下このパターンのクラスを拡張していく．

**定義 2 ($\mathcal{DSP}_0$)** パターン $P$ が以下の条件を満たすとき $\mathcal{DSP}_0$ であるという．(1) 2 階のパターンである．(2) パターン中の自由変数 $p$ の引数 $E_1, \ldots, E_m$ は次の条件を満たす．

i. $E_i$ は自由変数を持つ．

ii. q を自由変数とするとき，任意の文脈 $D$ について次の判定は成り立たない．
$\phi \vdash \lambda \bar{x}.\, q\ E_1 \cdots E_{i-1}\ E_{i+1} \cdots E_m \to \lambda \bar{x}.\, D[E_i]$

iii. $E_i$ 中の自由変数は $P$ 中では自由ではない． □

このパターンは文献 [6] の定義 1 の ii の部分の条件を緩めたものである．例えば，$\lambda x\, y.\, p\ \ x\ \ (y\ \ x)$ は $\mathcal{DSP}_0$ である．(2-ii) は自由変数の引数は互いに他の引数を用いても構成されないことを表している．すなわち，$E_i$ で構成される項は，他の引数の項 $E_1, \ldots, E_{i-1}, E_{i+1}, \ldots, E_m$ をどのように用いても構成できないのである．直観的にいうと，このことから決定性が生じている．$\mathcal{DSP}_0$ に属するパターン中の自由変数の引数の条件を緩め次のような $\mathcal{DSP}_n$ を再帰的に定義する．

**定義 3 ($\mathcal{DSP}_n$)** $\mathcal{DSP}_0$ は定義 2の通り．$\mathcal{DSP}_i$ $(0 \le i \le n-1)$ のいずれかに属する $\lambda \bar{x}.\, E_j$ $(1 \le j \le m)$ を考える．次の条件を満たすとき $p$(fresh) が自由となる項 $\lambda \bar{x}.\, p\ E_1 \cdots E_m$ は $\mathcal{DSP}_n$ である．

1. 2 階のパターンである．

$$
\begin{array}{l}
\mathit{discharge}\ s\ c\ =\ c \\
\mathit{discharge}\ s\ v\ =\ \mathit{replace}\ s\ v \\
\mathit{discharge}\ s\ (\lambda x.\ T_1)\ = \\
\quad \textbf{let}\ T' = \mathit{replace}\ s\ (\lambda x.\ T_1) \\
\quad \textbf{in if}\ T' = (\lambda x.\ T_1) \\
\qquad \textbf{then}\ \lambda x.\ (\mathit{replace}\ s\ T_1) \\
\qquad \textbf{else}\ T' \\
\mathit{discharge}\ s\ (T_1\ T_2)\ = \\
\quad \textbf{let}\ T' = \mathit{replace}\ s\ (T_1\ T_2) \\
\quad \textbf{in if}\ T' = (T_1\ T_2) \\
\qquad \textbf{then}\ ((\mathit{discharge}\ s\ T_1)\ (\mathit{discharge}\ s\ T_2)) \\
\qquad \textbf{else}\ T' \\
\mathit{replace}\ [\,]\ T_1\ =\ T_1 \\
\mathit{replace}\ ((E, T') : s)\ T_1\ = \\
\qquad \textbf{if}\ T' = T_1\ \textbf{then}\ E\ \textbf{else}\ \mathit{replace}\ s\ T_1
\end{array}
$$

図 2: Discharging Algorithm

2. q を自由変数とするとき，任意の文脈 $D$ について次の判定は成り立たない．
$\phi \vdash \lambda\bar{x}.q\ E_1 \cdots E_{i-1}\ E_{i+1} \cdots E_m \sim \lambda\bar{x}.D[E_i]$

□

パターンのクラス $\mathcal{DSP}$ を次のように定義する．

**定義 4** ($\mathcal{DSP}$) $\mathcal{DSP} \equiv \bigcup_{n=0}^{\infty} \mathcal{DSP}_n$ □

特に，線形の $\mathcal{DSP}_n$ を $\mathcal{LDSP}_n$ といい，線形の $\mathcal{DSP}$ を $\mathcal{LDSP}$ という．

**定義 5 (Discharge)** 項 $T$ の部分項 $T_1$ を図 2のように定義される関数 $\mathit{discharge}$ によって $\mathit{discharge}\ (x, T_1)\ T$ というように置き換えるとき $T$ から $T_1$ を $x$ で discharge するという． □

例えば，$p\ x\ (y\ x)$ から $y\ x$ を $y_1$ で discharge すると $p\ x\ y_1$ となる．

**補題 6 (マッチングの決定性)** すべての $i$ についてルール $X_i = \lambda\bar{x}.E_i \to \lambda\bar{x}.F_i$ が決定論的マッチを持つとき，ルール $Y = \lambda\bar{x}.c\ \bar{E} \to \lambda\bar{x}.d\ \bar{F}$ とルール $Z = \lambda\bar{x}.x_j\ \bar{E} \to \lambda\bar{x}.x_j\ \bar{F} (1 \le j \le m)$ は決定論的マッチを持つ．但し，$c, d$ は定数とし，$\bar{x}$ は空でも良いとする．

**証明** $\phi_0 \vdash \lambda\bar{x}.c \to \lambda\bar{x}.d$ は $c = d$ のとき $\{\phi_0 \mapsto \lambda x.x\}$ であり，$c \neq d$ のときマッチは存在しない．よって $\phi_0$ は決定論的に求まる．また仮定より $\phi_i \vdash X_i (1 \le i \le m)$ は決定論的に求まり，互いに compatible であるとき $\phi = \phi_0 . \cdots . \phi_m$ は決定論的に求まり，$\phi \vdash Y$ である．ルール $X$ についても，ルール $Y$ についてと同様の議論で証明できる． □

**補題 7 ($\mathcal{DSP}_0$ のマッチングの決定性)** パターンが $\mathcal{DSP}_0$ のマッチングは決定論的である．

**証明** パターンの上での帰納法を用いる．$P \to T$ というルールのマッチングを行うとする．パターンの形は一般に $\lambda\bar{x}.T'$ とかける．但し $\bar{x}$ は空であることもありえる．

**$T'$ が Rigid のとき** 項の body の頭が束縛された変数のときと定数のときがありえるが，同様に証明できるので，定数のときだけ示す．$P = \lambda\bar{x}.c$, $T = \lambda\bar{x}.d$ のとき，$c = d$ なら恒等写像が決定論的マッチである．$P$ の body が定数でないとき，$\eta$ 変換を用いれば，$P = \lambda\bar{x}.c\ \bar{E}$，$T = \lambda\bar{x}.d\ \bar{F}$ としても一般性を失わない．$\bar{E}$ のと $\bar{F}$ の個数が合わないときマッチは得られない．よって $\bar{E}$ のと $\bar{F}$ の個数が等しいときのことを考える．ルール $P \to T$ から得られるマッチは，次のルール群から得られるマッチの合成である．

$$
\begin{array}{l}
c \to d \\
\forall i.\ \lambda\bar{x}.E_i \to \lambda\bar{x}.F_i
\end{array}
$$

定義 2より，$c \in \mathcal{DSP}_0\ \wedge\ \lambda\bar{x}.E_i \in \mathcal{DSP}_0$ である．よって，帰納法の仮定よりこれらのルールのマッチングは決定論的である．また，補題 6よりルール $P \to T$ のマッチはこれらのルールから得られるマッチの合成から得られるマッチであり，決定論的に求まる．

**$T'$ が flexible のとき** $P = \lambda\bar{x}.p$, $T = \lambda\bar{x}.T'$ のとき，得られる可能性のあるマッチは $\{p \mapsto T'\}$ のただ 1 つである．そうでないとき，$\eta$ 変換を用いれば，$P = \lambda\bar{x}.p\ \bar{E}$，$T = \lambda\bar{x}.T'$ としても一般性を失わない．最汎マッチを得るためには，$T'$ の中の $E_i$ の出現をすべて discharge しなくてはならない．なぜなら，もし $E_i$ の出現が残っていたら得られるマッチの値域には $x \in FV(E_i)$ が自由に出現してしまうからである．また，定義 2の 1 より，$E_i$ は $\lambda$ 抽象ではないので，discharge は syntactical な置き換えで実現される．ここで，自由変数の引数の間に $E_i \trianglelefteq E_j (i \neq j)$ という包

含関係があったとする．そして $T'$ として $D[E_j]$ を考える．このとき $D[E_j]$ を $E_j$ より先に $E_i$ で discharge したものは $E_j$ では discharge できなくなるが，定義 2の 2 の ii によって，

$$\neg\exists C, D.\ C[E_1, \ldots, E_{j-1}, E_{j+1}, \ldots, E_m] =_{\alpha\beta\eta} D[E_j]$$

であるので，$D[E_j]$ は $E_j$ 以外の引数を用いても構成できない．即ち，$D[E_j]$ は $E_1, \ldots, E_{j-1}, E_{j+1}, \ldots, E_m$ を用いてどのように discharge しようとも自由変数が残ってしまう．よってすべての自由変数を取り除くような discharge をするためには，$E_i$ の包含関係の極大のものから discharge していく必要がある．また，定義 2の 1 により $E_i \not\trianglelefteq E_j \wedge E_j \not\trianglelefteq E_i (i \neq j)$ という包含関係のない 2 つの引数は互いに重なるところを discharge できないので discharge の適用順序に影響を与えない．

以上による唯一に決まる discharge の仕方によって，このマッチングは決定的である． □

**定理 8 (*DSP* のマッチングの決定性)** *パターンが DSP のマッチングは決定論的である．*

**証明** パターンの上での帰納法を用いる．

パターンが $\mathcal{DSP}_0$ であるとき補題 7 よりマッチングは決定論的である．

パターンが $\mathcal{DSP}_{n+1} (n \geq 0)$ であるときを考える．パターンが rigid であるとき，補題 7の rigid での議論と同様にして証明できる．よってパターンが flexible であるときのみを考える．$P = \lambda\bar{x}.p\ \bar{E}$，$T = \lambda\bar{x}.T'$ としても一般性を失わない．今，$Dom(\psi) = FV(P) - \{p\}$ という条件を満たす置換 $\psi$ を考える．$\psi\ E_i$ は $\lambda\bar{x}$ に束縛される自由変数を含むので，最汎マッチを得るためには，$T'$ の中の $\psi\ E_i$ の出現をすべて取り除く必要がある．なぜなら $T'$ の中の自由変数が discharge されずに残るとマッチが得られないからである．

以下は，補題 7の flexible での議論とほぼ同じ流れで証明できる．$\psi$ が決定論的に求まったとすると，$\psi$ が決定論的である必要条件を求めこれが定義より導けることを示す．定義 3の 1 より，$E_i$ は $\lambda$ 抽象ではないので，discharge は文法的な置き換えで実現される．ここで，自由変数の引数の間に $\psi\ E_i \trianglelefteq \psi\ E_j (i \neq j)$ という包含関係があったとする．そして $T'$ として $D[\psi\ E_j]$ を考える．このとき $D[\psi\ E_j]$ を $\psi\ E_j$ より先に $\psi\ E_i$ で discharge したものは $\psi\ E_j$ では discharge できなくなるが，定義 3の 2 によって，

$$\neg\exists C, D.\ C[\psi\, E_1, \ldots, \psi\, E_{j-1}, \psi\, E_{j+1}, \ldots, \psi\, E_m] =_{\alpha\beta\eta} D[\psi\, E_j]$$

であるので，$D[\psi\, E_j]$ は $\psi\, E_j$ 以外の引数を用いても構成できない．即ち，$D[\psi\, E_j]$ は $\psi\, E_1, \ldots, \psi\, E_{j-1}, \psi\, E_{j+1}, \ldots, \psi\, E_m$ を用いてどのように discharge しようとも自由変数が残ってしまう．よってすべての自由変数を取り除くような discharge をするためには，$\psi\, E_i$ の包含関係の極大のものから discharge していく必要がある．また，定義 2の 1 により $E_i \not\trianglelefteq E_j \wedge E_j \not\trianglelefteq E_i (i \neq j)$ という包含関係のない 2 つの引数は互いに重なるところを discharge できないので discharge の適用順序に影響を与えない．

次に，$\psi$ が決定論的に求まることを示す．$\psi$ が決定論的である必要条件は，discharge するときに，関数 *replace* によって部分項が discharge する項に置き換えるための条件を満たすことである．今，置き換える部分項を $T_1$ とするとこの条件は $\psi \vdash \lambda\bar{x}.E_i \to \lambda\bar{x}.T_1$ と書ける．定義 3より $\lambda\bar{x}.E_i$ は $\mathcal{DSP}_n$ である．帰納法の仮定によりパターンが $\mathcal{DSP}_n$ であるマッチングは決定論的であるので，$\psi$ は決定論的に求まる．

この唯一に決まる discharge の仕方によって，このマッチングは決定論的である． □

一般に，決定論的 2 階のパターンは必ずしも $\mathcal{DSP}$ ではない．例えば，$p$ が 2 回出現する非線形なパターン $\lambda x.p\ (p\ x) (\notin \mathcal{DSP})$ は決定論的 2 階のパターンである．しかし，線形のパターンに限れば決定論的 2 階のパターンは次の定理で見るように必ず $\mathcal{DSP}$ である．

**定理 9 ($\mathcal{LDSP}$ の完全性)** *線形で決定論的 2 階パターンは $\mathcal{LDSP}$ である．*

**証明** 今，考えているパターンは 2 階のパターンなので $\forall i.ord(E_i) = 1$ である．

まず，自由変数の引数の部分式の中に自由変数が出てこないときを考える．このようなパターンは一般に文脈 $C$ を用いて，$\lambda\bar{x}.C[p\ \bar{E}, \bar{q}, \bar{x}]$ と書ける．仮定よりパターンは線形であるので，$\lambda\bar{x}.C[E_i, \bar{c}, \bar{x}]$ という項とのマッチは少なくとも

$$\{p \mapsto \lambda\bar{y}.y_i\} \cup \psi,$$
$$\{p \mapsto \lambda\bar{y}.E_i\} \cup \psi$$

の 2 種類が得られ決定論的ではない．よってパターンの中の自由変数の引数は自由変数を含んでいなくてはならない．また，次のような置換が存在したとする．

$$\phi \vdash \lambda\bar{x}.\, r\ E_1 \cdots E_{i-1}\ E_{i+1} \cdots E_m \to \lambda\bar{x}.\, E_i \quad (1)$$

仮定よりパターンは線形であるので，項 $\lambda\bar{x}.\, C[E_i, \bar{c}, \bar{x}]$ とのマッチは少なくとも

$$\{p \mapsto \lambda\bar{y}.\, y_i\} \cup \psi,$$
$$\{p \mapsto \lambda\bar{y}.\, (\phi\ r)\ y_1 \cdots y_{i-1}\ y_{i+1} \cdots y_m\} \cup \psi$$

の 2 種類が得られ決定論的ではない．よって式 (1) を満たす $\phi$ は存在してはならない．以上より自由変数の引数の部分式の中に自由変数が出てこない線形で決定論的パターンは $\mathcal{DSP}_0$ である．

次に，自由変数の引数の部分式の中に自由変数が出てくるときを考える．次のような単一化子が存在したとする．

$$\phi \vdash \lambda\bar{x}.\, r\ E_1 \cdots E_{i-1}\ E_{i+1} \cdots E_m \sim \lambda\bar{x}.\, E_i \quad (2)$$

$\lambda\bar{x}.\, E_i$ 中の自由変数を定義域にもつ置換 $\psi$ を用いて表される次のルールを考える．

$$\lambda\bar{x}.\, C[p\ \bar{E}, \bar{q}, \bar{x}] \to \lambda\bar{x}.\, C[\psi\ E_i, \bar{c}, \bar{x}]$$

仮定よりパターンは線形なので，このルールを満たすマッチは少なくとも

$$\{p \mapsto \lambda\bar{x}.\, x_i\} \cup \psi$$
$$\{p \mapsto \lambda\bar{x}.\, (\phi\ r)\ x_1 \cdots x_{i-1}\ x_{i+1} \cdots x_m\} \cup \psi$$

の 2 種類が存在する．よって式 (2) を満たす単一化子が存在してはならない．したがって自由変数の引数の部分式の中に自由変数が出てくる線形で決定論的パターンは $\mathcal{DSP}_n$ である．

以上より線形で決定論的 2 階パターンは $\mathcal{LDSP}$ であることが証明された． □

## 3 応用

2 階の決定論的パターン $\mathcal{DSP}$ の定義中には自由変数の引数の間にマッチが存在するかどうかの判定が使われていた．実際のプログラム変換では，構文的な条件だけで定められるクラスが望ましい．実際のプログラム変換で十分な記述力をもち，かつ構文的な条件だけで定められるクラスは文献 [6] を参照して欲しい．

$\mathcal{DSP}$ の例として次のようなパターンがある．

$$\lambda f\ x.\, \textbf{if}\ p\ x\ \textbf{then}\ q\ x\ \textbf{else}\ r(b\ x, f(c\ x))$$

但し，$b, c$ は定数，$p, q, r$ は変数である．$\mathcal{DSP}$ は幅広い関数を表すことができる．実際，プログラム変換システム MAG[3](ver.2.1) のプログラム変換規則のパターンの内 Miller のパターンは 71.0%であるが $\mathcal{DSP}$ は 90.7% である．

## 4 まとめと今後の課題

本論文では決定論的線形 2 階パターンの必要十分条件を示した．この必要十分条件は直観的にわかりやすいものなので，プログラム変換システムの実装すれば，ユーザが理解しやすい高階パターンマッチングを提供することが出来る．また，一般の 2 階マッチングは NP 完全であるので効率が悪いが，われわれの提案するクラスのパターンについては項の大きさに対して線形であるので効率が良い．本論文の成果を，単一化や，3 階以上のマッチングに応用することと，プログラム変換システムにおける応用の検討や，関数プログラムのパターンを拡張することは今後の課題である．

### 参考文献


[1] Baxter, L.: *The complexity of unification*, PhD Thesis, Department of Computer Science, University of Waterloo, 1977.

[2] Bruckner, B., Liu, J., Shi, H., and Wolff, B.: Towards Correct, Efficient and Reusable Transformational Developments, *KORSO: Methods, Languages, and Tools for Construction of Correct Software*, LNCS, Vol. 1009, Springer-Verlag, 1995, pp. 270–184.

[3] de Moor, O. and Sittampalam, G.: Generic Program Transformation, *Proc. of the 3rd International Summer School on Advanced Functional Programming*, LNCS, Vol. 1608, Braga, Portugal, Springer-Verlag, September 1998, pp. 116–149.

[4] Huet, G. and Lang, B.: Proving and Applying Program Transformations Expressed with Second-Order Patterns, *Acta Informatica*, Vol. 11(1978), pp. 31–55.

[5] Miller, D.: A logic programming language with lambda abstraction. function variables, and simple unification, *Journal of Computer and System Sciences*, Vol. 1, 1991, pp. 479–536.

[6] Yokoyama, T., Hu, Z., and Takeichi, M.: Deterministic Second-order Patterns, *International Symposium on Logic-based Program Synthesis and Transformation*, Uppsalla, Sewden, August 2003. to appear.